

After initializing the surface and coils, the core part is calling minization algorithms to optimize coil parameters.

[called by: [focus](#).]

[calls: [bnormal](#), [bmnharm](#), [torflux](#), [length](#), [coilsep](#), [descent](#), [congrad](#), [hybridnt](#), [truncnt](#).]

Cost functions

The chi-squared optimization method is used here. The single target function is composed of several chosen object functions with user-supplied weights. The general formula is

$$\chi^2(\mathbf{X}) = \sum_j w_j \left(\frac{f_j(\mathbf{X}) - f_{j,o}}{f_{j,o}} \right)^2. \quad (1)$$

Currently, we have implemented constraints on Bnormal, Bmn harmonics, toroidal flux, coil length, coil-coil separation. For details, please view the documentation of each constraint.

Normalization

Besides the normalization terms in each constraint, like $|\mathbf{B}|$ in Bnormal, there is also an option to normalize the object function values to its initial value.

When **IsNormWeight = 1**, all the nonzero weights will be divided by the current object function values. For example, in the beginning, the Bnormal error is $f_{B_0} = 0.1$ and input $w_B = 1.0$. Then the updated $w'_B = w_B / f_{B_0} = 10.0$, such that at every step

$$w'_B f_B = w_B \frac{f_B}{f_{B_0}}. \quad (2)$$

** Please note that when writing the output file, the original weights (as same as input) and **IsNormWeight=1** are stored. So when you restart, the updated weights could be different.*

solvers.h last modified on Jun 26 09:34:37 2017 ;

[Focus subroutines](#);